

```
+++ date = '2025-02-21T10:18:14-08:00' draft = false title = 'Practica 0: Markdown, Git & Hugo' summary = "Jennifer Salgado Pacheco" +++
```

Primera sesión

Sintaxis y uso de markdown. Crear un reporte de qué es Markdown, cómo se utiliza y su sintaxis

Introducción

En esta primera sesión se hizo una breve introducción a Markdown. Aquí, se presentan los diferentes elementos que se pueden utilizar en Markdown, junto con una breve explicación de su propósito y uso.

Los encabezados se utilizan para crear títulos de diferentes niveles. Los encabezados varían según el número de #. Por ejemplo, # es el encabezado más grande, y ##### es el más pequeño.

Encabezado 1

Encabezado 2

Encabezado 3

Encabezado 4

Encabezado 5

Los itálicas se utilizan los símbolos _ o * para poner el texto en cursiva. Ejemplo:

Este es un texto en *itálica*

Este es un texto en *itálica*

Se utilizan ** o __ para poner el texto en negrita. Ejemplo:

Este es un texto en **negritas**

Este es un texto en **negritas**

En el tachado, se usa ~~ para crear texto con una línea tachada o rayada. Ejemplo:

Este es un texto ~~tachado~~

Listas no ordenadas (UL): Usan * o - para crear elementos de lista sin orden específico.

- Elemento 1
- Elemento 2
- Elemento 3

- Elemento 3.1
 - Elemento 3.2
 - Elemento 3.2.1
 - Elemento 4
-

Listas ordenadas (OL): Usan números seguidos de un punto (1., 2., etc.) para crear listas numeradas.

1. Elemento 1
 2. Elemento 2
 3. Elemento 3
 1. Elemento 3.1
 2. Elemento 3.2
 4. Elemento 4
-


Los enlaces se usan para crear hipervínculos. Puedes añadir un título al enlace usando comillas. Ejemplo:

[Texto de enlace](#)

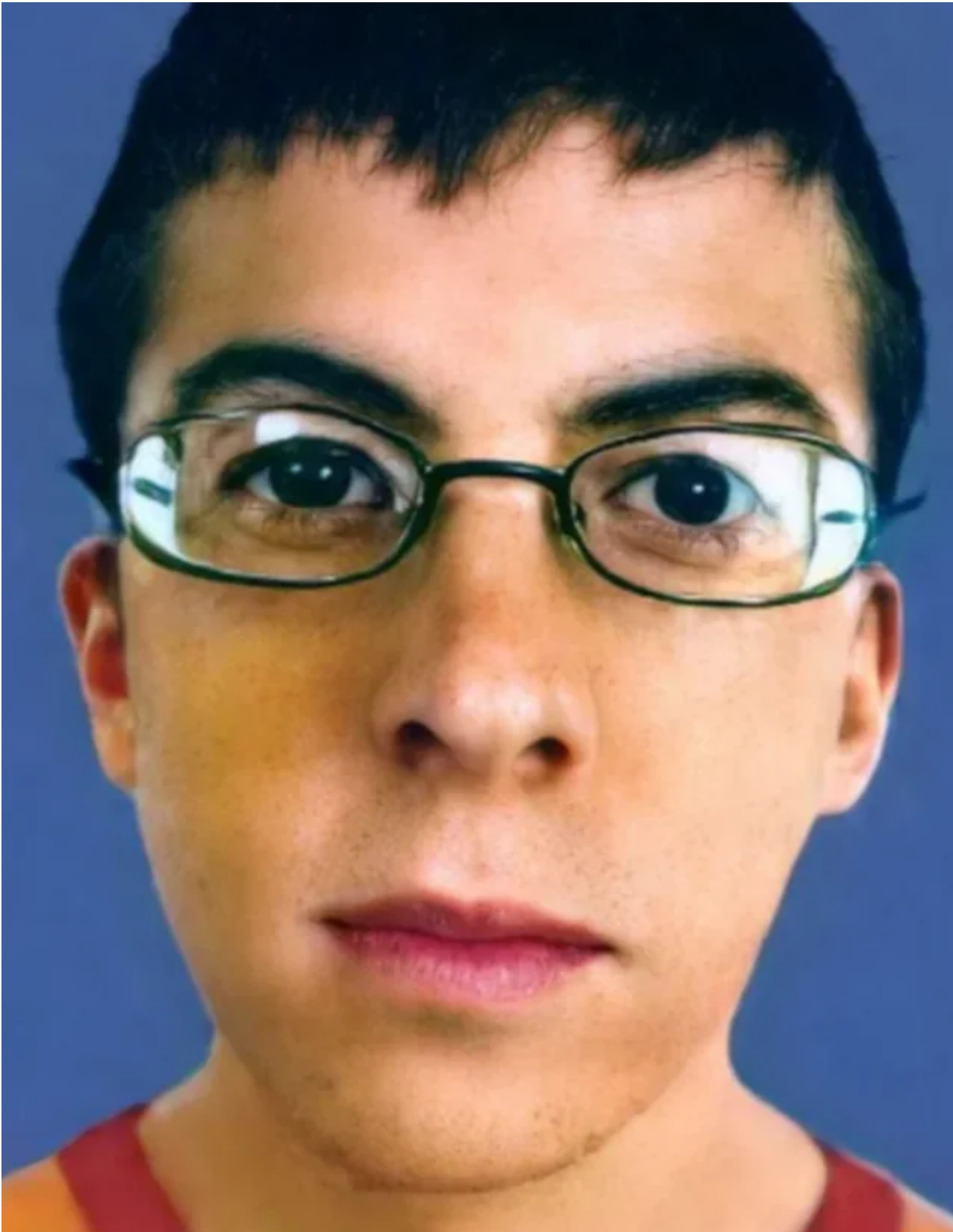
[UABC](#)

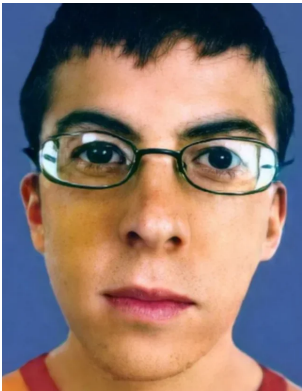
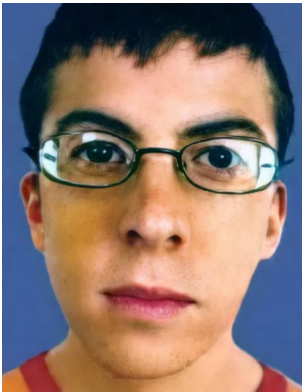
[UABC](#)

La sección de imágenes se usan para agregar imágenes. La diferencia con los enlaces es que se agrega un signo de exclamación (!) antes del texto.

 Texto alternativo







En los bloques de código, se usan las tres comillas invertidas (```) para crear bloques de código.

```
This is un code block
This is the second line of the code block
```

```
print("Hola mundo!")
```

```
console.log('Hello world!')

const test = ()
```

```
<h1>Hola mundo!</h1>
```

Para las tablas, se usa | para crear tablas y - para indicar los encabezados de cada columna.

Productos	Precio	Cantidad
Laptop	3.3	2
Mouse	13.3	1

Productos	Precio	Cantidad
Laptop	3.3	2
Mouse	13.3	1

☐ indica una tarea pendiente y [x] indica una tarea completada.

Esto es una nota

- ☒ Primera tarea
- ☐ Segunda tarea
- ☒ Tercera tarea
- ☐ Cuarta tarea

Los divisores horizontales se utilizan para separar secciones del texto con una línea horizontal. Se utiliza tirple *, - o _ para utilizarlos.

Para las menciones, puedes usar el símbolo @ seguido de un nombre de usuario para mencionar a alguien, junto con emoticonos (😄, 👍) para añadir reacciones. @tamagotchi0_0 👍 😄

Segunda sesión

Uso de Git y GitHub. Añadir al reporte que es Git y Github, como se utiliza, comandos esenciales para usar Git y subir la información a Github, como crear un repositorio y suben su información a la nube.

Git es un sistema de control de versiones distribuido que permite a los desarrolladores gestionar y rastrear cambios en el código fuente de manera eficiente. Fue creado por Linus Torvalds en 2005 y es ampliamente utilizado en la industria del software para la colaboración en proyectos de desarrollo. **GitHub** es una plataforma en línea que permite alojar repositorios Git y facilita la colaboración entre desarrolladores. Ofrece herramientas para el control de versiones, gestión de proyectos y automatización de tareas mediante integraciones con otros servicios.

Comandos esenciales de Git:

git init: Inicializa un repositorio Git en un directorio.

git clone URL_DEL_REPOSITORIO: Clona un repositorio existente desde una URL.

git status: Muestra el estado de los archivos en el repositorio.

git add nombre_archivo: Agrega un archivo específico al área de preparación.

git add .: Agrega todos los archivos modificados al área de preparación.

git commit -m "Mensaje descriptivo del cambio": Guarda los cambios en el historial con un mensaje descriptivo.

git log: Muestra el historial de cambios en el repositorio.

git pull origin main: Actualiza el repositorio local con los cambios remotos.

git push origin main: Envía los cambios confirmados al repositorio remoto.



Tercera sesión

Combinar los conocimientos de Markdown y Git para generar páginas estáticas con Hugo y GitHub Actions. Añadir al reporte que es Hugo y Github Actions, como crear un sitio estático en hugo, como suben esta información a Github, como configurar Github Actions para publicar el sitio en Github Pages.

En esta sesión, aprendimos a utilizar Hugo para la creación de nuestra página web, empleando el tema Ananke. En este sitio, publicaremos todas nuestras prácticas y reportes. Para optimizar su gestión, implementamos GitHub Actions, lo que nos permite automatizar el proceso y realizar un seguimiento de las diferentes versiones de la página.



Enlaces

[Mi repositorio Pagina estatica](#)